

**METHOD AND SYSTEM FOR PREDICTING  
UNDERSTANDING ERRORS IN AUTOMATED DIALOG  
SYSTEMS**

5

**Technical Field**

The invention relates to automated systems for communication recognition and understanding.

10 **Background Of The Invention**

Today there are many automated dialog systems in operation that serve many purposes, such as for customer care. Because they limit human involvement, such systems save millions of dollars in labor costs. Examples of such systems are shown in U.S. Patent Nos. 5,675,707, 5,860,063, 6,044,337, 15 and 6,173,261, and U.S. Patent Application Nos. 08/943,944, filed October 3, 1997, 09/699,494, 09/699,495, and 09/699,496 all filed October 31 2000, and 09/712,192 and 09/712,194, both filed November 15, 2000, each of which is incorporated by reference herein in its entirety.

While it has recently become possible to build spoken dialog systems that 20 interact with users in real-time in a range of domains, systems that support conversational natural language are still subject to a large number of language understanding errors. Endowing such systems with the ability to reliably distinguish language understanding errors from correctly understood communications might allow them to correct some errors automatically or to 25 interact with users to repair them, thereby improving the system's overall performance.

**Summary Of The Invention**

This invention concerns a method and system for monitoring an automated dialog system for the automatic recognition of language 30 understanding errors based on a user's input communications. The method may include determining whether a probability of understanding the user's input

communication exceeds a first threshold. The method may further operate such that if the first threshold is exceeded, further dialog is conducted with the user. Otherwise, the user may be directed to a human for assistance.

In another possible embodiment, the method operates as above except  
 5 that if the probability exceeds a second threshold, the second threshold being higher than the first, then further dialog is conducted with the user using the current dialog strategy. However, if the probability falls between a first threshold and a second threshold, the dialog strategy may be adapted in order to improve the chances of conducting a successful dialog with the user.

10 This process may be cumulative. In particular, the first dialog exchange may be stored in a database. Then, a second dialog exchange is conducted with the user. As a result, a second determination is made as to whether the user's input communication can be understood can be conducted based on the stored first exchange and the current second exchanges. This cumulative process may  
 15 continue using a third and fourth exchange, if necessary.

#### **Brief Description of the Drawings**

The invention is described in detail with reference to the following drawings wherein like numerals reference like elements, and wherein:

Fig. 1 is a block diagram of an exemplary NLU monitoring system;

20 Fig. 2 is a flowchart illustrating an exemplary NLU monitoring process;

Fig. 3 is a flowchart illustrating another exemplary NLU monitoring process using dialog strategy adaptation techniques;

Fig. 4 is a block diagram of an exemplary task classification system using a NLU monitoring system;

25 Fig. 5 is a flowchart illustrating an exemplary task classification process using a NLU monitoring process; and

Fig. 6 is a flowchart illustrating an exemplary task classification process using a NLU monitoring process having dialog strategy adaptation techniques.

#### **Detailed DescriptionOf The Preferred Embodiments**

30 Automated dialog systems promise efficient and natural access to a large variety of information sources and services from any communication device, such

as telephones, cellular or mobile phones, computers, videophones, etc. Several hundred systems that support system-initiative dialogs are currently being field-tested and deployed. However, systems that support mixed-initiative, conversational, natural language interaction are still subject to a large number of language understanding errors, which have a large impact on the system's performance.

This invention concerns monitoring an automated dialog system for the automatic recognition of language understanding errors. The ability to recognize and predict language understanding errors will endow an automated system with the ability to correct some errors automatically or to interact with users to repair them.

As an example of the benefit of error detection, contrast the exemplary dialogs below between human users and first, a conventional dialog system, and then a dialog system including the features of the present invention (the phone numbers, card numbers, and pin numbers in the sample dialogs are artificial.):

Dialog failure (failure to complete the user's intended task)

S1: AT&T How may I help you?

U1: *(silence)*

S2: Sorry. Please briefly tell me how I may help you?

U2: I wanta find out how much my bill's worth.

S3: What phone number would you like the call billed to?

U3: *Um* the access number.

S4: Sorry. Could I have that phone number again?

U4: 1 8 hundred 2 2 5 5 2 8 8.

S5: What number would you like to call?

U5: 800 225 5288 *(touchtone input)*

S6: Thank you.

The conventional dialog system frequently does not detect that an NLU error has occurred. The conventional dialog system misunderstands the user's

utterance in U2 as a request to make a third-number call. The system continues the dialog with utterance S3 by asking which number the user would like to bill the call to. The dialog ends without completing the user's task.

5     Sample successful dialog

S1: AT&T How may I help you?

U1: I need to [ uh ] put a call on my calling card please

S2: May I have your card number, please?

U2: 7 6 5 4 3 2 1 0 9 8 7 6 5 4

10    S3: What number would you like to call?

U3: 8 1 4 7 7 7 6 6 6 6 (*misunderstood*)

S4: May I have that number again?

U4: 8 1 4 7 7 7 6 6 6 6

S5: Thank you.

15

In contrast, the system of the invention completed the above dialog successfully. Note that after detecting the NLU error in utterance U3, the system of the invention re-prompts the user to repeat the calling card number in utterance S4.

20

While the above example illustrates the invention in a telephone customer care system, this invention may be applied to any single mode, or multimodal, dialog system. Furthermore, the invention may apply to any automated recognition and understanding system that receives communications from external sources, such as users, customers, service providers, associates, etc.

25

Consequently, the method may operate in conjunction with one or more communication networks, including a telephone network, the Internet, an intranet, Cable TV network, a local area network (LAN), a wireless communication network, etc.

30

In the discussion below, the term "natural language understanding" is used to describe the understanding of any form of communications received over the networks which may be expressed verbally, nonverbally, multimodally, etc.

Examples of nonverbal communications include the use of gestures, body movements, head movements, non-responses, text, keyboard entries, keypad entries, mouse clicks, DTMF codes, pointers, stylus, cable set-top box entries, graphical user interface entries, touchscreen entries, etc. Multimodal

5 communications involve communications on a plurality of channels, such as aural, visual, etc. However, for ease of discussion, examples and discussions of the method and system of the invention will be discussed below in relation to telephone systems.

Fig. 1 illustrates an exemplary natural language understanding monitoring  
10 system 100. The natural language understanding monitoring system 100 includes a natural language understanding (NLU) monitor 180, a dialog manager 190, a training database 165 and a dialog history database 170. The NLU monitor 180 receives recognition data from a recognizer 120 and understanding data from a natural language understanding (NLU) unit 130 that are based on  
15 input communications from the user.

The recognizer 120 and the NLU unit 130 are shown as separate units for clarification purposes. However, the functions of the recognizer 120 and the NLU unit 130 may be performed by a single unit within the spirit and scope of this invention.

20 The recognizer 120 and the NLU unit 130 may operate using one or more of a variety of recognition and understanding algorithms. For example, the recognizer 120 and the NLU unit 130 may use confidence functions to determine whether the user's input communications have been recognized and understood. The recognition and understanding data from the user's input communication are  
25 also input into the NLU monitor 180. Based on this data, the NLU monitor 180 calculates a probability that the language is understood clearly and this may be used in conjunction with other mechanisms like recognition confidence scores to decide whether and/or how to further process the user's communication.

As a result, if the user's input communication can be satisfactorily  
30 recognized and understood, the NLU unit 130 routes and/or processes the user's input communication, which may include the request, comment, etc. However, if

the NLU monitor 180 recognizes errors in the understanding of the user's input communication such that it cannot be satisfactorily recognized and understood, dialog with the user may need to be conducted. This process will be described in greater detail below.

5           In the natural language understanding monitoring system 100, the dialog history database 170 serves as a database for storing each dialog exchange for a particular dialog. The training database 165 stores NLU errors collected from interactions with human users and models built based on those errors, the NLU features identified from the collected dialogs, and the NLU rules generated from  
10           the dialogs and the NLU features. The NLU monitor 180 exploits the training database 165 by using the dialog history stored in the dialog history database 170 to predict whether a NLU error is to occur in the current dialog. While the training database 165 and the dialog history database 170 are shown as separate databases in the exemplary embodiments, the dialog history and  
15           training data may be stored in the same database or memory, for example. This database or memory may be stored external or internal to the system.

          As discussed below, the NLU monitor 180 of the natural language understanding monitoring system 100 can be trained to improve its ability to detect errors by exploiting language understanding errors collected in  
20           interactions with human users and stored in the training database 165. The initial segments of these dialogs can be used to predict that a problem is likely to occur. The ability to predict language understanding errors will allow the system's dialog manager 190 to apply more sophisticated strategies to repairing problems, and even perhaps, to prevent them.

25           Note that the recognizer 120 may be trained to recognize any number of communication symbols, both acoustic and non-acoustic, including grammar fragments, meaningful words, meaningful phrases, meaningful phrase clusters, superwords, morphemes, multimodal signals, etc., using any of the methods known to one skilled in the art including those found in U.S. Patent Nos.  
30           5,675,707, 5,860,063 and 6,044,337, and U.S. Patent Application Nos. 08/943,944, 09/712,192 and 09/712,194, as discussed above.

In describing the invention, three classes of NLU outcomes are distinguished: RCORRECT, a correctly understood utterance; RPARTIAL-MATCH, a partially understood utterance; and RMISMATCH a misunderstood utterance. Experiments were conducted on learning to automatically distinguish these three classes of NLU outcomes in 11,787 spoken utterances collected in a field trials system interacting with live customer traffic.

During the trial, the behaviors of all the system's 100 components were automatically recorded in a log file and the dialogs were later transcribed by humans and labeled with one or more of the categories representing the intended result/task that the caller was asking the system 100 to perform, on a per communication basis. This label will be referred to as the HUMAN LABEL.

The NLU unit 130 also logged what it believed to be the correct task category. This label will be referred to as the NLU LABEL. One of the focuses of this invention is on the problem of improving the system's 100 ability to automatically detect when the NLU LABEL is wrong. As mentioned above, this ability would allow the system 100 to make better decisions about when to transfer to a human customer care agent, but it might also support repairing such misunderstandings, either automatically or by interacting with a human caller.

The experiments reported here primarily utilize a rule learning program to automatically induce an NLU error classification model from the 11,787 utterances in the corpus. While other learning methods may be used within the spirit and scope of the invention, the experiments and examples discussed below utilized if-then rules that are used to express the learned classification model. For this purpose, if-then rules are easy to understand and would affect the ease with which the learned rules could be integrated back into the system 100.

In this classification, the names of a set of *classes* to be learned, the names and ranges of values of a fixed set of *features*, and *training data* specifying the class and feature values for each example in a training set, were input. As a result, a classification model for predicting the class of future examples was output. For exemplary purposes, the classification model is

learned using greedy search guided by an information gain metric, and is expressed as an ordered set of if-then rules.

In application, the utterances in the corpus must be encoded in terms of a set of classes (the output classification) and a set of input features that are used as predictors for the classes. As mentioned above, three classes are distinguished based on comparing the NLU LABEL, the HUMAN LABEL and recognition results for card and telephone numbers: (1) RCORRECT: NLU correctly identified the task and any digit strings were also correctly recognized; (2) RPARTIAL-MATCH: NLU correctly recognized the task but there was an error in recognizing a calling card number or a phone number; (2) RMISMATCH: NLU did not correctly identify the user's task. The RCORRECT class accounts for 7481 (63.47%) of the utterances in the corpus. The RPARTIAL-MATCH accounts for 109 (0.1%) of the utterances, and the RMISMATCH class accounts for 4197 (35.6%) of the utterances.

Next, each utterance is encoded in terms of a set of 43 features that have the potential to be used during runtime to alter the course of the dialog. These features were either derived from these features. The system 100 components that we extracted features from were the recognizer 120, the NLU unit 130, and the dialog manager 190, along with a representation of the discourse history.

Because the contribution of different system 100 components to the problem of predicting NLU error are to be examined, a classifier that had access to all the features is trained and its performance is compared to classifiers that only had access to recognizer 120 features, to NLU unit 130 features, and to discourse contextual features. Below we describe features obtained from each module:

#### Recognizer Features

- recog, recog-numwords, asr-duration, dtmf-flag, rg-modality, rg-grammar, tempo

#### NLU Unit Features



- a confidence measure for all of the possible tasks that the user could be trying to do
- salience-coverage, inconsistency, context-shift, top-task, nexttop-task, top-confidence, diff-confidence, confpertime, salpertime

5

#### Dialog Manager and Discourse History Features

- sys-LABEL, utt-id, prompt, reprompt, confirmation, sub-dial,
- discourse history: num-reprompts, num-confirms, num-subdials, reprompt%, confirmation%, subdialog%

10

The recognizer 120 takes the user's input communication and produces a transcription. The recognizer 120 features extracted from the corpus were the output of the recognizer 120 (*recog*), the number of words in the recognizer 120 output (*recog-numwords*), the duration in seconds of the input to the recognizer 120 (*asr-duration*), a flag for touchtone input (*dtmf-flag*), the input modality expected by the recognizer 120 (*rg-modality*) (one of: none, speech, touchtone, speech+touchtone, touchtone-card, speech+touchtone-card, touchtone-date, speech+touchtone-date, or none-final-prompt), and the grammar used by the recognizer 120 (*rg-grammar*). A feature called *tempo* was also calculated by dividing the value of the *asr-duration* feature by the *recog-numwords* feature.

15

20

The motivation for the recognizer 120 features is that any one of them may have impacted recognition performance with a concomitant effect on language understanding. For example, *asr-duration* has consistently been found to correlate with incorrect recognition. The name of the grammar (*rg-grammar*) could also be a predictor of NLU errors since it is well known that the larger the grammar, the more likely a recognizer 120 error is to occur.

25

One motivation for the *tempo* feature is that it has been found that users tend to slow down their communication (speech, movement, etc.) when the system 100 has misunderstood them. This strategy actually leads to more errors since the recognizer 120 is not trained on this type of communication. The *tempo* feature may also indicate hesitations, pauses, or interruptions, which

30

could also lead to recognizer 120 errors. On the other hand, additional multimodal input such as touchtone (DTMF) in combination with the user's communication, as encoded by the feature *dtmf-flag*, might increase the likelihood of understanding, since the touchtone input is unambiguous it can

5 constrain language understanding.

The goal of the NLU unit 130 module is to identify which of the possible results, tasks, etc. the user is attempting, and extract from the user's communication any items of information that are relevant to completing the user's intended result (e.g. a phone number is needed for the task *dial for me*). In this

10 example, fifteen of the features from the NLU unit 130 represent the distribution for each of the 15 possible tasks of the NLU unit's 130 confidence in its belief that the user is attempting that task. Also included is a feature to represent which task has the highest confidence score (*top-task*), and which task has the second highest confidence score (*nexttop-task*), as well as the value of the highest

15 confidence score (*top-confidence*), and the difference in values between the top and next-to-top confidence scores (*diff-confidence*).

Other features represent other aspects of the NLU unit 130 processing of the utterance. The *inconsistency* feature is an intra-utterance measure of semantic diversity, according to a task model of the domain. Some task classes

20 occur together quite naturally within a single statement or request. For example, the *dial for me* task is compatible with the *collect call* task, but is not compatible with the *billing credit* task.

The *salience-coverage* feature measures the proportion of the utterance that is covered by the salient grammar fragments. For example, this may include

25 the whole of a phone or card number if it occurs within a fragment. The *context-shift* feature is an inter-utterance measure of the extent of a shift of context away from the current task focus, caused by the appearance of salient phrases that are incompatible with it, according to a task model of the domain.

In addition, similar to the way we calculated the *tempo* feature, the

30 *salience-coverage* and *top-confidence* features are normalized by dividing them by *asr-duration* to produce the *salpertime* and *confpertime* features. The

motivation for these NLU features is to make use of information that the NLU unit 130 has as a result of processing the output of recognizer 120 and the current discourse context. For example, for utterances that follow the first utterance, the NLU unit 130 knows what task it believes the user is trying to complete. The

5 *context-shift* feature incorporates this knowledge of the discourse history, with the motivation that if it appears that the caller has changed his/her mind, then the NLU unit 130 may have misunderstood an utterance.

The function of the dialog manager 190 is to take as input the output of the NLU unit 130, decide what task the user is trying to accomplish, decide what the

10 system will say next, and update the discourse history. The dialog manager 190 decides whether it believes there is a single unambiguous task that the user is trying to accomplish, and how to resolve any ambiguity.

Features based on information that the dialog manager 190 logged about its decisions or features representing the on-going history of the dialog might also

15 be useful predictors of NLU errors. Some of the potentially interesting dialog manager 190 events arise due to low NLU confidence levels computed by the NLU monitor 180 that lead the dialog manager 190 to *re-prompt* the user or *confirm* its understanding. A re-prompt might be a variant of the same question that was asked before, or it could include asking the user to choose between two

20 tasks that have been assigned similar confidences by the NLU monitor 180. For example, in the dialog below the system utterance in S3 counts as a re-prompt because it is a variant of the question in utterance S2.

S1: AT&T How may I help you?

25 U1: I need credit please.

S2: Okay. What is the reason for the credit?

U2: Miss, uh, Miss, different area code than I needed.

S3: Sorry. I need to know whether you need credit for a wrong number, bad connection or a call that was cut off.

30 (*re-prompt*)

U3: It is a wrong number.

S4: What was the number that you dialed?

The features that are extracted from the dialog manager 190 are the task-type label, *sys-label*, whose set of values include a value to indicate when the system had insufficient information to decide on a specific task-type, the utterance id within the dialog (*utt-id*), the name of the prompt played to the user (*prompt*), and whether the type of prompt was a re-prompt (*reprompt*), a confirmation (*confirm*), or a subdialog prompt (a superset of the re-prompts and confirmation prompts (*subdial*)). The *sys-label* feature is intended to capture the fact that some tasks may be harder than others. The *utt-id* feature is motivated by the idea that the length of the dialog may be important, possibly in combination with other features like *sys-label*. The different prompt features again are motivated by results indicating that re-prompts are frustrating for users.

The discourse history features included running tallies for the number of re-prompts (*num-reprompts*), number of confirmation prompts (*num-confirms*), and number of subdialog prompts (*num-subdials*), that had been played before the utterance currently being processed, as well as running percentages (*percent-reprompts*, *percent-confirms*, *percent-subdials*). The use of running tallies and percentages is based on showing that normalized features are more likely to produce generalized predictors.

Fig. 2 is a flowchart of an exemplary natural language understanding monitoring process. The process begins its step 2000 and goes to step 2100 where the NLU monitor 180 receives recognition and understanding data from the recognizer 120 and the NLU unit 130, respectively, based on an input communication from a user. In step 2200, the NLU monitor 180 determines the probability of whether the user's input communication maybe correctly understood based on the initial dialog exchange data based on the recognition and understanding data received and the decision rules stored in the training database 165.

If the probability of correctly understanding the user's input communication determined by the NLU monitor 180 does not exceed a predetermined threshold,

for example, the NLU monitor 180 signals the dialog manager 190 to route the user to a human for assistance. This situation may be represented by the following relationship:

$$t1 > P(\text{understanding})$$

- 5            If the probability of correctly understanding the user's input communication determined by the NLU monitor 180 exceeds the predetermined threshold, for example, the NLU monitor 180 believes that continued dialog may be conducted with the user which may result in successfully routing and/or processing the user's input communication. This situation may be represented by the following relationship:
- 10           relationship:

$$t1 < P(\text{understanding})$$

As result, the process moves to step 2400 where the NLU monitor 180 stores the dialog exchange into dialog history database 170.

- 15           In step 2500, the dialog manager 190 conducts further dialog with the user to obtain clarification of the users initial input communication (exchange 1). In step 2600, if the recognizer 120 and NLU unit 130 can recognize and understand the user's second input communication (exchange 2) so that it can be processed, the process goes to step 2700 and ends. However, if after exchange 2 the user's input communication cannot be understood and processed, the process returns
- 20           to step 2100 where the recognition and understanding data from the user's second input communication is input to the NLU monitor 180 by the recognizer 120 and the NLU 130. Then in step 2200, the NLU monitor 180 determines the probability of correctly understanding the user's input communication based on both the current exchange 2 and previous exchange 1 retrieved from the dialog
- 25           history database 165. The process that continues as stated above.

- Fig. 3 is a flowchart of an exemplary natural language understanding monitoring process that allows the dialog strategy to be adapted between dialog exchanges. Steps 3000, 3100, 3200, 3300, and 3400, operate similarly to steps 2000, 2100, 2200, 2300, and 2400 of Fig. 2, respectively, and will not be further
- 30           discussed.

However, after step 3400 in which the first dialog exchange is stored in the dialog history database 170, in step 3500, the NLU monitor 180 determines whether the probability of correctly understanding the user's input communication exceeds a second threshold. If the NLU monitor 180 determines that probability of probability of correctly understanding the user's input communication exceeds the second threshold, in step 3600, the dialog manager 190 conducts dialog with the user using the current dialog strategy. This situation may be represented by the following relationship:

$$t1 < P(\text{understanding}) > t2$$

However, if the NLU monitor 180 determines that the probability of correctly understanding the user's input communication does not exceed the second threshold, in step 3700, the dialog manager 190 conducts dialog with the user using an adapted (or different) dialog strategy with the user. This situation may be represented by the following relationship:

$$t1 < P(\text{understanding}) < t2$$

The process then proceeds to step 3800, where determination is made whether the user's input communication can be processed based on either the current dialog strategy or the adapted dialog strategy opted for in step 3500. The process that continues similar to Fig. 2 above, and as such, will not be discussed further.

Fig. 4 illustrates one possible application of the natural language understanding monitoring system 100. In particular, Fig. 4 shows an exemplary automated task classification system 400 that conducts automated dialog with a user. The automated task classification system 400 includes a classification subsystem 410, a labeled training communications database 450 and a natural language understanding monitoring system 100. The classification subsystem 410 includes a recognizer 420, an NLU unit 430 and a task classification processor 440. The natural language understanding monitoring system 100 includes an NLU monitor 180, a dialog manager 190, a training database 165 and a dialog history database 170, as discussed above.

The automated task classification system 400 is based on the notion of task classification and routing. In the automated task classification system 400, services that the user can access are classified into a plurality of categories by a task classification processor or alternatively, to a category called *other* for calls  
 5 that cannot be automated and must be transferred to a human operator. Each category describes a different task, such as person-to-person dialing, or receiving credit for a misdialed number.

The task classification processor 440 determines which task the user is requesting on the basis of the NLU unit's 430 understanding of recognized  
 10 language segments, stored in the labeled training communications database 450, and recognized by the recognizer 420. The NLU 430 attempts to interpret the user's response to, for example, the open-ended system greeting *AT&T, How May I Help You?* Once the response has been recognized and understood, task classification processor 440 determines the task, and the information needed for  
 15 completing the caller's request is obtained using the dialog manager 190. The dialog manager 190 uses sub-modules that are specific for each task.

While the task classification processor 440 and the NLU unit are shown as separate units in Fig. 4, their functions may be performed by a single unit within the spirit and scope of the invention.

20 Fig. 5 is a flowchart of a possible automated task classification process using the natural language understanding monitoring system 100 of the invention. Process begins its step 5000 in proceeds to step 5100 where an input communication is received by the recognizer 420. At step 5200, the recognizer 420 attempts to recognize portions of the user's input communication, including  
 25 grammar fragments, meaningful words/phrases/symbols, morphemes, actions, gestures, or any other communication signal.

At step 5300, based on the understanding received from the NLU unit 430, the NLU monitor 180 determines whether the user's input communication can be correctly understood so that the task can be classified. If the task can be  
 30 classified, then in step 5400, the task classification processor 440 routes the

user's request according to the task classification, and the process proceeds to step 5900 and ends.

However, in step 5300, if the NLU monitor 180 determines that the task classification processor 440 cannot classify the user's request, in step 5500, the NLU monitor 180 determines whether the probability of correctly understanding the user's input communication exists above a predetermined threshold. In this iteration, the NLU monitor 180 is using only the first exchange (exchange 1). The NLU monitor 180 uses the classification model stored in the dialog training database 165 to determine whether the probability of correctly understanding the user's input communication exceeds the predetermined threshold. If the NLU monitor 180 determines that the probability of correctly understanding the user's input communication does not exceed the threshold, then in step 5800 for user is routed to a human for assistance. The process then goes to step 5900 and ends.

If, in step 5500, the NLU monitor 180 determines that the probability of correctly understanding the user's input communication is above the predetermined threshold based on the stored classification model, in step 5600 the first dialog exchange is stored in the dialog history database 170. Then, in step 5700, the dialog manager 190 conducts dialog with the user to obtain further clarification of user's request. The process returns to step 5200 wherein the recognizer 420 attempts to recognize portions of the second exchange with the user. Steps 5300 and 5400 are performed as before using the second exchange.

However, in step 5400, if the task classification processor 140 determines that the task cannot be classified, the dialog predictor 180 gauges the probability of conducting successful dialog above the predetermined threshold based on both exchange 2 and exchange 1 stored in the dialog history database 170. If the probability of conducting a successful dialog does not exceed the predetermined threshold based on exchanges 1 and 2, in step 5800, the task classification processor 440 is instructed by the dialog manager 190 to route the user to a human assistant.



On the other hand, in step 5500, if the NLU monitor 180 determines that the probability of correctly understanding the user's input communication exceeds the predetermined threshold, the second dialog exchange may be stored in step 5600 and further dialog conducted with the user in step 5700. The process would continue using the current and previous exchanges with the user until the task is completed or the user is routed to a human assistant.

Fig. 6 is a flowchart of a possible automated task classification process using the natural language understanding monitoring system 100 with the ability to adapt dialog strategy in an effort to conduct successful dialog with the user. Steps 6000, 6050, 6100, 6150, 6250, 6300, 6350, and 6400 operate similarly to steps 5000, 5100, 5200, 5300, 5400, 5500, 5800, and 5600 of Fig. 5, respectively, and will not be discussed further.

However, after step 6400 in Fig. 6 in which the first dialog exchange is stored in the dialog history database 170, in step 6450, the NLU monitor 180 determines whether the probability of correctly understanding the user's input communication exceeds a second threshold. If the NLU monitor 180 determines that the probability of correctly understanding the user's input communication exceeds the second threshold, in step 6500, the dialog manager 190 conducts dialog with the user using the current dialog strategy.

However, if the NLU monitor 180 determines that the probability of correctly understanding the user's input communication does not exceed the second threshold, in step 6550, the dialog manager 190 conducts dialog with the user using an adapted (or different) dialog strategy with the user.

The process then proceeds to back to step 6050, where the recognizer 120 receives the user's input communication based on either the current dialog strategy or the adapted dialog strategy opted for in step 6450. The process that continues similar to Fig. 5 above, and as such, will not be discussed further.

The output of each experiment is a classification model learned from the training data that is stored in the training database 165. The model is evaluated in several ways. First, multiple models are trained using different feature sets extracted from different system 100 components in order to determine which

feature sets are having the largest impact on performance. Second, for each feature set, the error rates of the learned classification models are estimated using ten-fold cross-validation, by training on a random 10,608 utterances and testing on a random 1,179 utterances 10 successive times. Third, precision, recall and the confusion matrix are recorded the classifier trained on all the features tested on a random held-out 20% test set. Fourth, for the classifier trained on all the features, the extent to which the error can be minimized on the error classes RMISMATCH and RPARTIAL-MATCH is examined by manipulating the rule-learning program's loss ratio parameter. Finally, the results of training other learners are compared on the same dataset with several of the feature sets. The overall accuracy results for detecting NLU errors using the rule-learning program are summarized in Fig. 7 below (SE=Standard Error).

	<u>Features Used</u>	<u>Accuracy</u>	<u>(SE)</u>
15	BASELINE (majority class)	63.47 %	
	ALL	86.16 %	(0.38)
	NLU UNIT ONLY	84.80 %	(0.38)
	RECOGNIZER + DISCOURSE	80.97 %	(0.26)
	RECOGNIZER ONLY	78.89 %	(0.27)
20	DISCOURSE ONLY	71.97 %	(0.40)

The first line of the above table represents the accuracy from always guessing the majority class (RCORRECT); this is the BASELINE against which the other results should be compared. The first row labeled ALL, shows the accuracy based on using all the features available from the system 100 components. This classifier can identify NLU errors 23% better than the baseline. The second row of the table, NLU ONLY, shows that the classifier based only on the NLU unit 130 features performs statistically as well as the classifier based on all the features. The third row of the table, RECOGNIZER + DISCOURSE shows that combining the RECOGNIZER features with the DISCOURSE features produces a significant increase in accuracy over the use of

recognizer 120 features alone, which however still performs worse than the NLU unit 130 features on their own. The last two rows, RECOGNIZER ONLY and DISCOURSE ONLY, indicate that it is possible to do significantly better than the baseline using only the features from the recognizer 120 or from the dialog manager 190 and the discourse history, but these features on their own cannot do as well at predicting NLU accuracy as the NLU unit's 130 own features based on its own calculations.

Using ten-fold cross-validation, the resulting NLU error classifier can correctly identify whether an utterance is an NLU error 86% of the time, an improvement of 23% over the majority class baseline. In addition, the most important features are those that the NLU unit 130 can compute, suggesting that it will be straightforward to integrate the NLU monitor 180 into the NLU unit 130 of the system 100.

Further results are discussed in Walker et al., "Using Natural Language Processing and Discourse Features to Identify Understanding Errors in a Spoken Dialogue System", International Conference on Machine Learning, January 31, 2000, incorporated herein by reference in its entirety.

As shown in Figs. 1 and 4, the method of this invention may be implemented using one or more programmed processors. However, method can also be implemented on a general-purpose or a special purpose computer, a programmed microprocessor or microcontroller, peripheral integrated circuit elements, an application-specific integrated circuit (ASIC) or other integrated circuits, hardware/electronic logic circuits, such as a discrete element circuit, a programmable logic device, such as a PLD, PLA, FPGA, or PAL, or the like. In general, any device on which the finite state machine capable of implementing the flowcharts shown in Figs. 2, 3, 5 and 6 can be used to implement the task classification system and natural language understanding monitoring functions of this invention.

While the invention has been described with reference to the above embodiments, it is to be understood that these embodiments are purely exemplary in nature. Thus, the invention is not restricted to the particular forms

shown in the foregoing embodiments. Various modifications and alterations can be made thereto without departing from the spirit and scope of the invention.